

Remotely accessible laboratory model for enabling improved understanding of robot simultaneous localisation and mapping

Bruce Moulton & Caesar Natividad

University of Technology, Sydney
Sydney, Australia

ABSTRACT: The University of Technology Sydney (UTS) remotely accessible laboratory system started in 2001, and allows staff and students to access apparatus including microcontrollers, programmable logic controllers, FPGAs, coupled water tanks and loaded beams. This article describes the preliminary stages of the development of a new remotely accessible component that is specifically tailored for UTS students of robotics. Remotely accessible robots will be used by students of at least two subjects, Mechatronics 2 and Advanced Robotics. New learning activities are being designed to introduce students to the laser range finder and camera, enable students to gain an understanding of how instrumentation data is logged and used to create and compare maps, and to facilitate experiments with localisation and path planning. It is envisaged that by using the remotely accessible robotics laboratories, students will be able to devote a greater proportion of their time to furthering their understanding of key robotics concepts. It is proposed that remotely accessible laboratories have a potential role in enabling student's understanding of such key concepts.

INTRODUCTION

In recent years, there has been an emerging trend towards the use of remotely accessible laboratories for the purpose of enabling various learning outcomes. This paper explores the role of such laboratories in enabling experience with mobile robots. It is proposed that remotely accessible laboratories may be suitable for enabling specialist skills relating to experiments with robot control and using data from sensors to test map-building and localisation algorithms.

Robotic equipment has traditionally been relatively expensive to purchase and maintain. This presents challenges when attempting to make such equipment available for large classes of students. Several institutions have implemented solutions where students are able to access/control/program robots through a remotely located server.

The first industrial robot that could be controlled from anywhere in the world using an internet browser is thought to be a University of Western Australia *Telelabs* robot. It has been estimated that since 1994 approximately 500,000 users from 100 different countries have used the robot to stack blocks [1].

An early remotely accessible laboratory, which allowed a robot to be programmed from a remote location was installed at the University of Essex from 2001 [2]. The system used a Pioneer robot, and did not require a local copy of any software on the user's PC. To program the robot from a remote location, a user wrote a script in COLBERT (a language for Saphira), and used a Web form to submit the script. The system allowed students to control and program the robot from a remote location.

Carusi et al describe a remote robotic laboratory system implemented at the University of Siena. It is based on the LEGO Mindstorm, and appears to focus more on robot control than on autonomous map building. Students using this lab focus on concepts relating to defining a mathematical algorithm to get the robot to follow a reference path. Path planning and obstacle avoidance experiments were planned for further development. The system was developed using Automatic Control Telelab, a remote laboratory system that had been in place at the University of Sienna since 1999. The interfaces use existing software environments such as LabVIEW and Matlab/Simulink, and the authors expressed the opinion that from a student-learning experience, such tools were preferable to *ad-hoc* specific, tailored environments [3].

A system used at the University of Craiova allows students to develop their own localisation algorithms [4]. The system is designed to teach students concepts in localisation, map building and path planning. A feature of the system is an emphasis on demonstrations, which are available to introduce students to the remote lab. It allows students to manually control the robot and observe the robot's feature recognition processes. Remotely located

students can also develop and update algorithms. The robot uses Microsoft Visual studio, which enables a student, once logged in, to debug and edit the code. It has become clear in testing phases of this robot that the design of the maze must be unique in each position to allow the robot to localize, and it must also be designed in a way to prevent the robot from getting stuck.

A robotic system used at FernUniversität, Hagen uses a remotely accessible Pioneer 3 AT mobile robot with a laser scanner [5]. Students experiment with the apparatus to enable skills related to behaviour based programming. A local copy of the Saphira (used to communicate with the Pioneer robot) program is required on the user's PC, and the remotely located copy of Saphira communicates with the robot over the Internet.

The UTS remotely accessible laboratory system started in 2001. It allows staff and students to access apparatus including microcontrollers, PLCs, FPGAs, coupled water tanks and loaded beams. Instead of remote control, it allows the remotely located systems to be *programmed*. Early versions of the UTS remote laboratory system are described in [6][7]. Efforts are currently being undertaken to develop new apparatus that is specifically tailored for UTS students of robotics.

METHOD/APPARATUS

The architecture of the UTS remotely accessible laboratories is summarised in Figure 1.

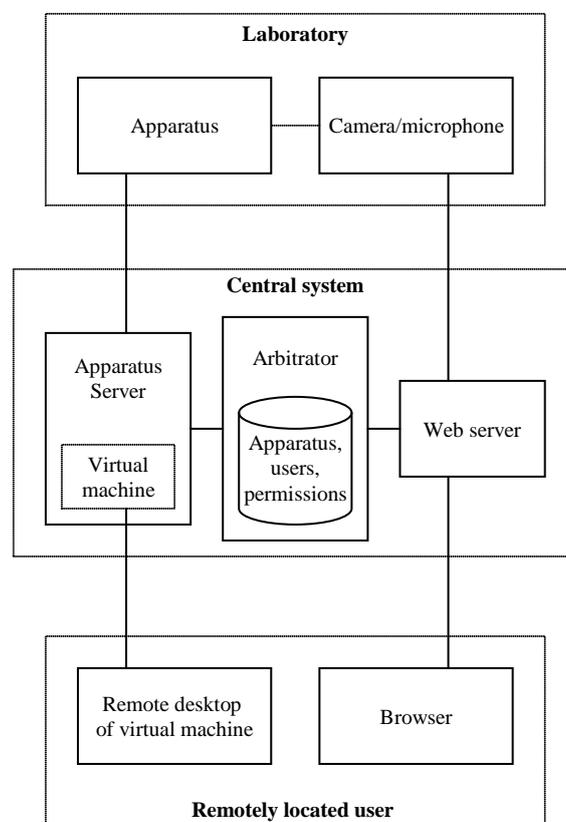


Figure 1: Overview of the UTS remotely accessible laboratories.

The user requests access to a piece of apparatus via the Web server. The Web server passes messages to and from the arbitrator, which checks if the user has the necessary permissions. The arbitrator places the user in a queue to wait until the apparatus is available. When the apparatus becomes available, a virtual machine is created on the apparatus server and associated with the appropriate apparatus. The student then establishes a remote desktop connection to the virtual machine. An interface with controls for the apparatus becomes accessible to the user via the remote desktop. Audio and visual is provided through the Web server.

One type of robot to be remotely accessible through the UTS system is an iRobot Create, a robust platform that supports serial communication of sensor data and actuation commands. This robot is supported by a free open source Player/Stage device interface abstraction layer.

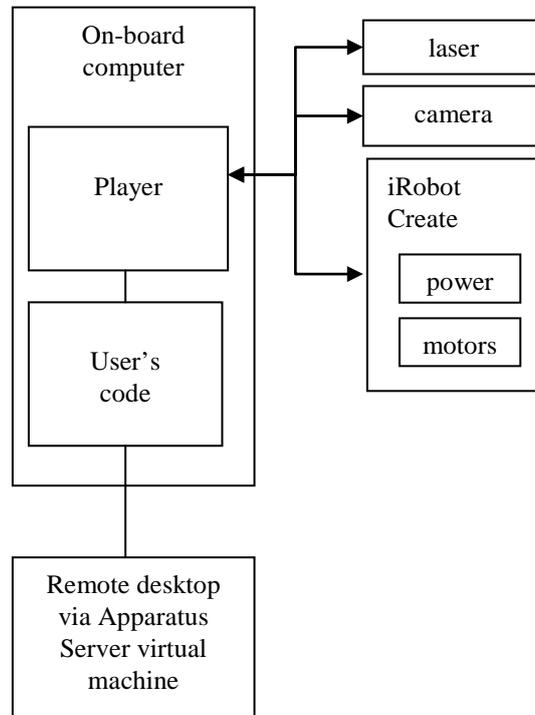


Figure 2: Overview of the robotic apparatus.

THE LEARNING ACTIVITIES

The remotely accessible robot will be used in two subjects, Mechatronics 2 and Advanced robotics. Some general objectives of the learning activities are to introduce students to robot sensors using the laser range finder and camera, enable students to gain an understanding of how instrumentation data is logged and used to create and compare maps, and to facilitate student experiments with localisation and path planning. The students will develop localisation algorithms offline and then test and validate the algorithms on the robot. Data (from laser range finder, camera and odometry) will be collected during autonomous operation and used to build maps of the maze, and to compare the characteristics of the different sources of data.

Students will gain understanding and experience of principles of localisation using the augmented Monte-Carlo Localisation algorithm (amcl). This algorithm is already built into Player. It is a particle filter and variation on Markov Localisation. It works by placing many possible poses on the map. It then ranks them by their probability that each is the correct pose by comparing the actual laser readings with the map readings at each possible pose. By using Bayes Rule and Gaussian probability, the possible poses become successively concentrated into the correct area until a single pose is chosen [8].

Students will learn about path planning using the wavefront algorithm. This algorithm works by dividing the configuration space into a grid cell pattern. It will then assign cost values to cells based on their proximity to an obstacle with free space value being 0. Then starting at the goal a cost value is calculated and placed in each adjacent cell based on its distance to the goal and added to the other cell value with the goal cell value equalling 0. The cells values are determined *rippling* out from the goal until the start cell has a value. By connecting cells with the lowest total cost value the shortest, safest route is planned. The algorithm within player will then break the plan down into straight lines and waypoints. The waypoints are then passed one at a time to the local navigator. The local navigator has obstacle avoidance planning. Alternative planners could include the visibility graph method or Voronoi diagram; however, these methods are not available within Player at the time of writing

Students will make use of two methods for obstacle avoidance/local navigation. These are the Nearness Diagram and Vector Field Histogram (VFH) drivers. The VFH method involves generating a polar histogram from the sensor and calculating a steering direction using the histogram. It is calculated by identifying the possible openings to fit the robot and then applies a cost function to each path. The cost function considers the direction of the goal relative to the each passage and the difference in angles between passages. The nearness diagram method is more suited to more cluttered spaces and thus has been chosen as the main method for the students to work with. The nearness diagram method creates similar histograms and makes judgments based on different cases, selecting the most promising direction of travel. Studies indicate that algorithms using nearness diagrams can be successfully used to navigate a circular robot in cluttered situation [9]. The obstacle avoidance algorithm will receive a waypoint from the path planner and use that as the goal, avoiding obstacles, and will receive the next waypoint upon arriving at the first.

Students ordinarily refer to the Player Manual [10] as part of their efforts to become familiar with the system. The manual gives an overview of the software and its uses. Links within the site lead to documentation on the drivers and other relevant information. The documentation includes information on augmented Monte Carlo localisation and other control algorithms and hardware drivers associated with the software

Figure 3 provides a simplified flowchart of a user's main function. It interfaces with the Player abstraction layer (It sends messages to and from Player, and Player sends messages to and from the robot.). The main loop simply alternates between two pre-defined goal locations (starting with goal1). Once it has reached the goal location it checks a Boolean variable, *Charge*. *Charge* is true if the battery goes below the threshold. If *Charge* is true, the goal location is set to goal3. Goal3 is a goal near the charging station. This will cause the robot to first to goal1, then goal2, then goal1 and so on, until the robot needs to charge. It will then move to goal3. Once at goal3 the robot will dock and charge the battery. After charging the robot will then proceed to goal1, and the cycle repeats.

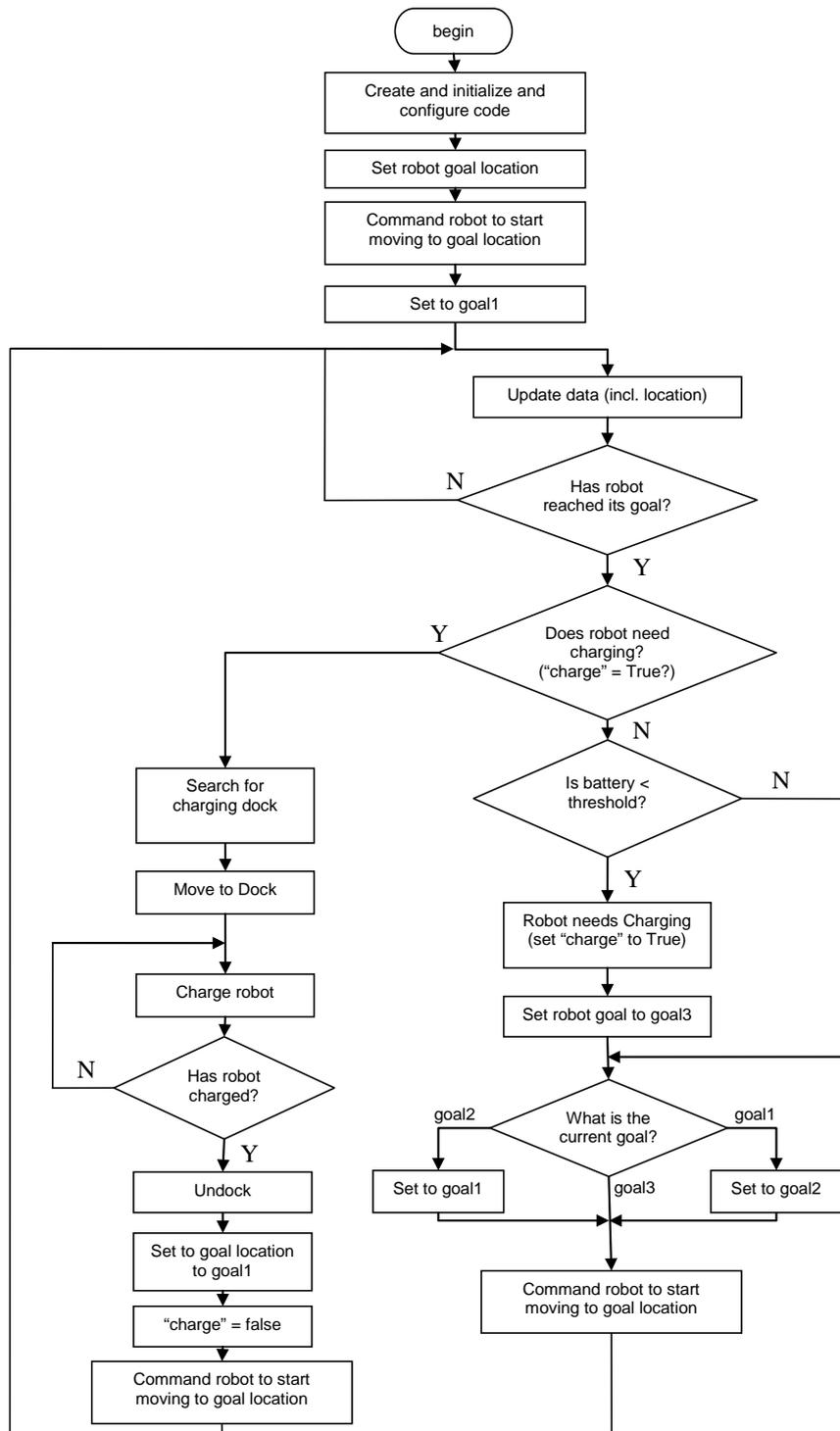


Figure 3: Simplified flowchart of a user's main function.

The student is not able to interact with the robot while it is autonomously exploring its maze. Students will only be able to observe the robot while data from various sensors are logged. The students will then separately analyse that data and compare data sources by building maps based on the data.

It would be possible to simply provide students with data and have them analyse it to produce a map. However, it is proposed that the remote lab will help in enabling students to develop an improved conceptual understanding of where the data has come from by giving them the opportunity to watch the source of the data in action. The system allows students to take as many sets of sample data as they wish. Students would typically analyse the data associated with each *run*, and make revisions to their code, while other students are accessing the apparatus.

REMOTELY ACCESSIBLE LABORATORIES ARE DIFFERENT FROM SIMULATED LABORATORIES

Distance learning laboratories have become more appealing in recent years largely because they have certain benefits over traditional proximal laboratories. Distance laboratories allow equal opportunities for students with difficulties accessing the equipment, and students are able to access the experiment at any time allowing more flexibility.

A common point of confusion arises regarding the distinction between a simulated (or virtual) laboratory and a remotely accessible laboratory. A virtual or simulated laboratory is ordinarily in essence a piece of software.

It is widely thought that simulations can aid in helping students to understand basic concepts. Studies suggest that simulated laboratories can enable conceptual understanding by reducing cognitive load, for example, because students do not need to adjust the physical equipment to get *sensible* data [11]. Another study suggests that simulated or virtual labs can be much more open ended than their real-world counterparts, and finds that simulated laboratories give students more freedom to experiment and test and examine their knowledge, rather than follow a procedure which is sometimes the case in proximal laboratories [12].

When attending traditional laboratories, a large proportion of a student's time is typically allocated to learning procedures, setting up apparatus, troubleshooting and cleaning apparatus. This means that less time is spent learning new concepts. However, the experience of troubleshooting can in itself be an important part of the student's learning.

Remotely accessible laboratories typically involve real *physical* apparatus. The fact that the apparatus is already set up, and the fact that it does not need to be maintained nor cleaned by the student, frees up a greater proportion of the student's time for learning the concepts that are essential for the curriculum. Hence, remotely accessible laboratories share many of the benefits of simulated laboratories. However, the fact that the apparatus is real, the accuracy of the sensors can drift, must be configured correctly, and things can go wrong, gives the students some opportunity to troubleshoot and to question the legitimacy of the data-important skills that are developed in physical laboratories but perhaps not developed by simulated laboratories. It may be the case that different forms of laboratories have different learning outcomes [13].

Students of robotics typically spend a large proportion of their laboratory time troubleshooting problems that are caused by faulty actuators, sensors or software. While the ability to deal with buggy software and hardware is a desirable graduate attribute, it is acknowledged that debugging can absorb large proportions of student time, and that some of this time might be better spent on the development of other skills and abilities. Hence, it is hoped that by using the remotely accessible robotics laboratory, students will be able to devote a greater proportion of their time to developing a greater understanding of key robotics concepts associated with mapping and localisation.

CONCLUSIONS

The UTS remotely accessible laboratories have been in operation since 2001, and new apparatus has been added to the system on a regular basis over the intervening years. This article overviews work undertaken to add apparatus which is intended to enable students to develop an improved understanding of robotics. The apparatus is intended to become available for students in the second half of 2009. The potential for the system in furthering student learning and understanding is discussed. Further work will be required to gain a greater understanding of the system's effects on robotics students' learning, and this work will be undertaken once the robotics component of system becomes operational.

REFERENCES

1. The Telelabs Project, University of Western Australia, 23 July 2009, <http://telerobot.mech.uwa.edu.au/index2.html>
2. Yu, L., Tsui, P., Zhou, Q. and Hu, H., A Web-based telerobotic system for research and education at Essex. *Proc. 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Como, Italy, 1, 37-42 (2001).
3. Carusi, F., Casini, M., Prattichizzo, D. and Vicino, A., Distance learning in robotics and automation by remote control of LEGO mobile robots. *Proc. IEEE International Conference on Robotics and Automation*, 2, 1,820-1, 825 (2004).

4. Dinulescu, I., Popescu, D. and Predescu, A., Remote learning environment for visual based robot navigation. *Proc. 19th EAEEIE Annual Conference*, Tallinn, Estonia, 26-30 (2008).
5. Borgolte, U., Interface design of a virtual laboratory for mobile robot programming. *Research, Reflections and Innovations in Integrating ICT in Education*, FORMATEX, Badajoz, Spain, 1, 402-406 (2009).
6. Moulton, B.D., Lasky, V.L. and Carmody, N.J., Practice-based engineering education: a distributed environment for teaching and learning embedded systems. *Proc. 5th UICEE Annual Conference on Engineering Education*, Chennai, India, 33-35 (2002).
7. Moulton, B.D., Lasky, V. L. and Murray, S.J., The development of a remote laboratory: educational issues. *World Trans. on Engng. and Technology Educ.*, 3, 1, 19-22 (2004).
8. Thrun, S., Fox D., Burgard W. and Dellaert, F., *Robust Monte Carlo Localization for Robots*. Elsevier Preprint, February (2001).
9. Minguez, J. and Montano, L., Nearness Diagram (ND) navigation: collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20, 1, 45-59 (2004).
10. Gerkey, B., Player Manual (2005), 20 March 2009, <http://playerstage.sourceforge.net/doc/Player-2.1.0/player>
11. Javidi, G. and Sheybani, E., Virtual engineering lab. *Proc. 36th Annual Frontiers in Education Conference*. San Diego, CA (2006).
12. Pyatt, K. and Sims, R., Learner performance and attitudes in traditional versus simulated laboratory experiences. In *ICT: Providing choices for learners and learning*. *Proc. ASCILITE*, Singapore (2007).
13. Lindsay, E.D. and Good, M.C., Effects of laboratory access modes upon learning outcomes. *IEEE Trans. on Educ.*, 48, 4, 619-631 (2005).